

(19) World Intellectual Property
Organization
International Bureau



(43) International Publication Date
6 May 2004 (06.05.2004)

PCT

(10) International Publication Number
WO 2004/038598 A1

(51) International Patent Classification⁷: **G06F 15/78**

(21) International Application Number:
PCT/EP2003/011176

(22) International Filing Date: 6 October 2003 (06.10.2003)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
0224627.0 23 October 2002 (23.10.2002) GB

(71) Applicant (for all designated States except US): **MOTOROLA INC** [US/US]; 1303 E.Algonquin Road, Schaumburg, IL 60196 (US).

(72) Inventor; and

(75) Inventor/Applicant (for US only): **RAUBUCH, Martin** [DE/DE]; Motorola GmbH, Halbleiter, Schatzbogen 7, 81829 Muenchen (DE).

(74) Agent: **HARRISON, Christopher**; Motorola European Intellectual, Property Operations, Midpoint, Alencon Link, Basingstoke, Hampshire RG21 7PL (GB).

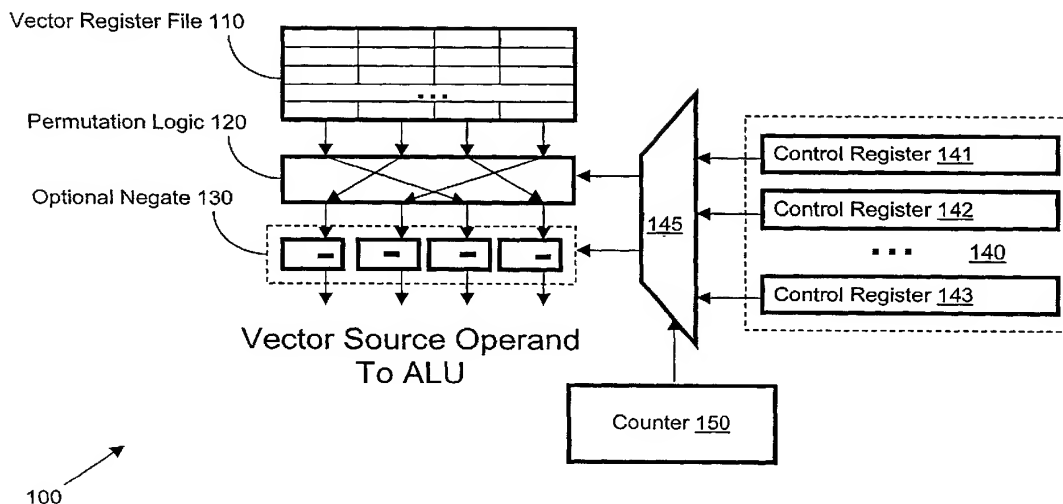
(81) Designated States (*national*): AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NI, NO, NZ, OM, PH, PL, PT, RO, RU, SC, SD, SE, SG, SK, SL, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, YU, ZA, ZM, ZW.

(84) Designated States (*regional*): ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HU, IE, IT, LU, MC, NL, PT, RO, SE, SI, SK, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:
— with international search report

[Continued on next page]

(54) Title: ARRANGEMENT, SYSTEM AND METHOD FOR VECTOR PERMUTATION IN SINGLE-INSTRUCTION MULTIPLE-DATA MICROPROCESSORS



(57) Abstract: A vector permutation system (100) for a single-instruction multiple-data microprocessor has a set of vector registers (110) which feed vectors to permutation logic (120) and then to a negate block (130) where they are permuted and selectively negated according to control parameters received from a selected one of a set of control registers (140). A control arrangement (145, 150) selects which control register is to provide the control parameters. In this way no separate permutation instructions are necessary or need to be executed, and no permutation parameters need to be stored in the vector registers (110). This leads to higher performance, a smaller vector registers file and hence a smaller size of the microprocessor and better program code density.

WO 2004/038598 A1



For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

- 1 -

ARRANGEMENT, SYSTEM AND METHOD FOR VECTOR PERMUTATION IN
SINGLE-INSTRUCTION MULTIPLE-DATA MICROPROCESSORS

5 **Field of the Invention**

This invention relates to microprocessors with Single-Instruction Multiple-Data (SIMD) capability.

10

Background of the Invention

In the field of this invention microprocessors with SIMD architecture are arranged to process vector operands. It
15 is known to provide instructions that permute (rearrange the order of) the components of vector operands in order to improve the efficiency of digital signal processing algorithms on SIMD microprocessors. Permutation parameters are required to determine the characteristics
20 of the permutation to be performed.

However, this approach has the disadvantage(s) that if the vector permutation requires extra instructions, performance decreases. If the permutation parameters
25 and/or the permuted vector operand require extra registers in the microprocessor's vector register file, a large register file is required. This increases the microprocessor's size and has a negative impact on program code density.

30

- 2 -

A need therefore exists for an arrangement, system and method for vector permutation in SIMD microprocessors wherein the abovementioned disadvantage(s) may be alleviated.

5

Statement of Invention

In accordance with a first aspect of the invention there
10 is provided an arrangement for vector permutation in SIMD microprocessors as claimed in claim 1.

In accordance with a first aspect of the invention there
is provided a system for vector permutation in SIMD
15 microprocessors as claimed in claim 2.

In accordance with a third aspect of the invention there
is provided a method for vector permutation in SIMD
microprocessors as claimed in claim 5.

20

The arrangement preferably further includes a negate
block coupled to the control means and coupled to receive
and selectively negate vectors from the permutation logic
block according to the control parameters received from
25 the control means, wherein the control parameters include
permutation parameters and negate parameters.

Preferably the control means includes at least one
counter arranged to provide a sequential order for
30 selecting one of the plurality of control registers.

- 3 -

The control register parameters are preferably also used for determining negate characteristics and the step of permutating further includes the step of selectively
5 negating the vectors according to the parameters of the selected control register. Preferably the step of selecting further includes the following of a sequential order of the plurality of control registers.

10 Preferably the sequential order includes automatic sequencing through a set of fixed control parameters. Alternatively the sequential order preferably includes automatic sequencing through a set of programmable control parameters. The sequential order is preferably
15 cyclical.

In this way an arrangement, system and method for vector permutation in SIMD microprocessors is provided in which no separate permutation instructions are necessary or
20 need to be executed, and no permutation parameters need be stored in the vector registers. This leads to higher performance, a smaller vector register file and hence a smaller size of the microprocessor and better program code density.

25

Brief Description of the Drawings

One arrangement, system and method for vector permutation
30 in SIMD microprocessors incorporating the present

- 4 -

invention will now be described, by way of example only, with reference to the accompanying drawings, in which:

FIG. 1 shows a block schematic diagram of a known
5 microprocessor with SIMD architecture; and

FIG. 2 shows a block schematic diagram of a
microprocessor system with SIMD architecture
incorporating the present invention.

10

Description of Preferred Embodiment(s)

Within the field of SIMD architecture, it is known that
15 permutation and optional negate operations of vector
operands may be performed as side operations of certain
instructions and do not themselves require separate
instructions or execution cycles.

20 However, programmers need control over when and how such
permutations are performed. In order to control when
permutations are performed, qualifiers are needed. These
qualifiers may be:

- enable/disable mechanisms
- 25 - vector register numbers
- instruction types
- other

In order to control how permutations are performed,
30 permutation parameters, source/destination operands or
optional negate operations are needed. Such permutation

- 5 -

parameters can either be fixed (hard-wired for specific algorithms) or programmable (stored in registers).

Referring now to FIG. 1, there is shown a prior art
5 microprocessor 5 with SIMD architecture. A vector
register file 10 of the microprocessor feeds vector
operands into a permutation logic block 20. The vector
register file 10 has a predetermined number of registers.
The number of general purpose and/or vector registers in
10 modern Reduced Instruction Set Chip (RISC) machines
typically is an integer to the power of 2 with 8/16/32/64
being the most common numbers. In the example depicted in
FIG. 1, there are 32 128-bit registers, each register
having four 32-bit elements. The last register (register
15 15) is used to store control parameters for controlling
the permutation logic block 20, as depicted by arrow 17.

Referring now to FIG. 2, there is shown a microprocessor.
100 with SIMD architecture. A read port of a vector
20 register file 110 feeds vector operands into a
permutation logic block 120 and from there into a negate
logic block 130. The vector register file 110 has a
predetermined number of registers. In the example
depicted in FIG. 2, there are 8 128-bit registers (of
25 which 5 are shown), each register having four 32-bit
elements.

The output is typically used as source operand for a
vector Arithmetic Logic Unit (ALU) (not shown).

- 6 -

Permutation and negate parameters relating to permutations to be performed upon the vectors of the vector register file 110 are stored as control parameters in a series of control registers 140. A control block 145
5 is coupled to each of the series of control registers 140 and is further coupled to provide the control parameters therefrom to control the permutation logic block 120 and the optional negate logic block 130. A counter 150 is also coupled to the control block 145, the counter 150
10 being arranged to determine which of the series of control registers is coupled via the control block 145 to the permutation logic block 120 and the optional negate logic block 130 at any one time.

15 In operation, the microprocessor 100 will commence with the counter 150 pointing at a given control register of the series 140, such as a first control register 141. When a permutation is to be performed (all qualifiers true), the control parameters (permutation and negate
20 parameters) stored in the first control register 141 are provided via the control block 145 to the permutation logic block 120 and to the optional negate logic block 130. The contents of the vector register file 110 are then processed by the permutation logic block 120 and the
25 optional negate logic block 130 according to these control parameters. It will be noted that the optional negate logic block 130, being optional, may or may not perform a negate function on the contents of the vector register file 110, depending upon the received control
30 parameters.

- 7 -

Once processed, the output vector source operand is sent to the ALU (not shown) and the counter 150 is incremented. This causes the control block 145 to select the next control register of the series 140 (such as the
5 second control register 142) for the next permutation. The counter 150 is arranged to cycle through each of the series of control registers 140 in a repeating manner.

It will be understood that the an arrangement, system and
10 method for vector permutation in SIMD microprocessors described above provides the following advantages:
No extra instructions are required to permute/negate the components of vector operands, leading to higher performance. Furthermore, no further registers of the
15 vector register file are required to store the permuted/negated vector operands and the permutation parameters. It should be noted that even with programmable permutation parameters, the control registers 140 of FIG. 2 are significantly smaller than
20 the vector register 15 of FIG. 1. Since the microprocessor's register file is smaller, this leads to a smaller size of the microprocessor and better program code density (fewer bits in op-codes for vector register addressing).

25

It will be appreciated by a person skilled in the art that alternative embodiments to that described above are possible. For example, the control register series 140 and counter 150 may be augmented by multiple counters and
30 control register series, coupled with qualifiers such as instruction type or register number. Also the counting

- 8 -

sequence need not repeat in a cyclical fashion, and it is possible to load the counter(s) with specific sequence start points by adding just one further instruction. All of these features may be used to add complexity to the
5 sequence of permutations and so further increase the flexibility of the architecture.

Furthermore the number and size of vector registers may differ from those described above, it being understood
10 that the number of vector registers required by the present invention will be less than that required for an equivalent prior art arrangement.

- 9 -

Claims

1. An arrangement for vector permutation in a single-instruction multiple-data microprocessor, the arrangement
5 comprising:
a permutation logic block coupled to receive and
permute vectors from at least one vector register
according to control parameters;
a plurality of control registers, each coupled to
10 selectively provide control parameters to the permutation
logic block; and,
control means coupled between the plurality of control
registers and the permutation logic block and arranged
for selecting one of the plurality of control registers
15 and for providing the control parameters from the
selected one of the plurality of control registers to the
permutation logic block.

2. A single-instruction multiple-data microprocessor
20 vector permutation system comprising:
at least one vector register;
a permutation logic block coupled to receive and
permute vectors from the at least one vector register
according to control parameters;
25 a plurality of control registers, each coupled to
selectively provide control parameters to the permutation
logic block; and,
control means coupled between the plurality of control
registers and the permutation logic block and arranged
30 for selecting one of the plurality of control registers
and for providing the control parameters from the

- 10 -

selected one of the plurality of control registers to the permutation logic block.

3. The arrangement of claim 1 or system of claim 2
5 further comprising a negate block coupled to the control means and coupled to receive and selectively negate vectors from the permutation logic block according to the control parameters received from the control means, wherein the control parameters include permutation
10 parameters and negate parameters.

4. The arrangement or system of any preceding claim wherein the control means includes at least one counter arranged to provide a sequential order for selecting one
15 of the plurality of control registers.

5. A method for vector permutation in a single-instruction multiple-data microprocessor, the method comprising the steps of:
20 providing vectors to be permuted;
selecting one of a plurality of control registers, each control register containing parameters for determining permutation characteristics;
permutating the vectors according to the parameters of
25 the selected control register.

6. The method of claim 5 wherein the control register parameters are also used for determining negate characteristics and the step of permutating further
30 includes the step of selectively negating the vectors

- 11 -

according to the parameters of the selected control register..

7. The method of claim 5 or claim 6 wherein the step of
5 selecting further includes the following of a sequential order of the plurality of control registers.

8. The arrangement or system of claim 4, or method of claim 7, wherein the sequential order includes automatic
10 sequencing through a set of fixed control parameters.

9. The arrangement or system of claim 4, or method of claim 7, wherein the sequential order includes automatic sequencing through a set of programmable control
15 parameters.

10. The arrangement, system or method of claims 4, 7, 8 or 9 wherein the sequential order is cyclical.

20 11. An arrangement for vector permutation in single-instruction multiple-data microprocessors substantially as hereinbefore described with reference to FIG. 2 of the accompanying drawings.

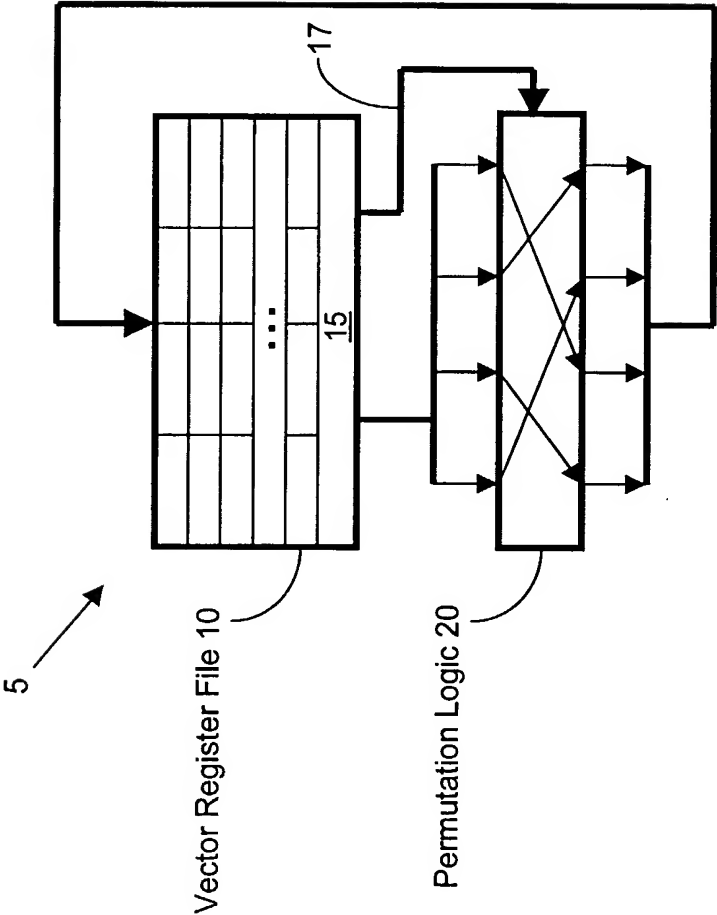
25 12. A system for vector permutation in single-instruction multiple-data microprocessors substantially as hereinbefore described with reference to FIG. 2 of the accompanying drawings.

30 13. A method for vector permutation in single-instruction multiple-data microprocessors substantially

- 12 -

as hereinbefore described with reference to FIG. 2 of the accompanying drawings.

FIG. 1
Prior Art



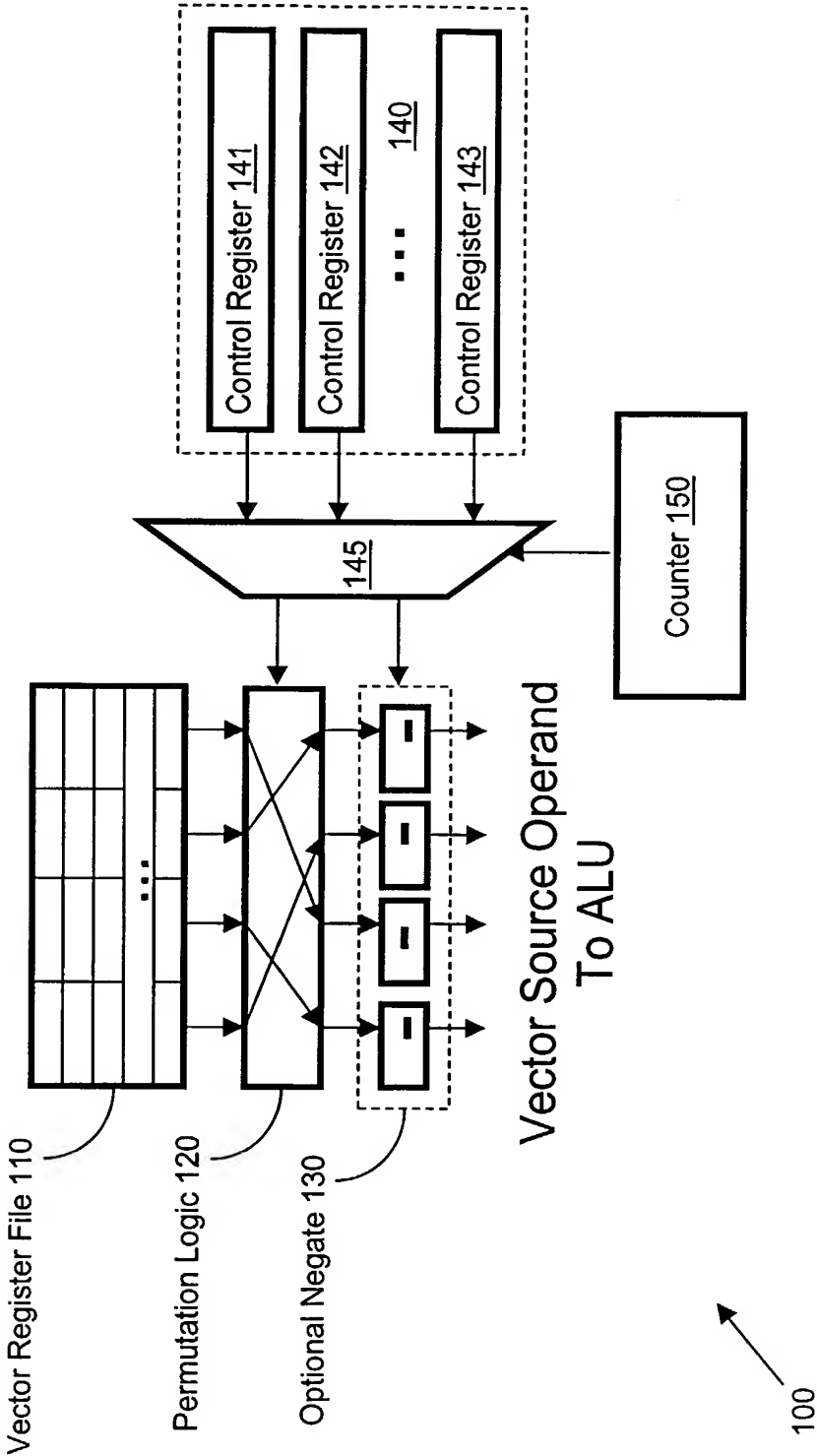


FIG. 2